

Gathering and Documenting Requirements

Case Study in Theoretical and Practical Considerations

A White Paper

CHARM

Utah Department of Health

Preparation of this publication was supported by a contract from
All Kids Count, a program of The Robert Wood Johnson Foundation.

Gathering and Documenting Requirements

Case Study in Theoretical and Practical Considerations

A White Paper

Rhoda Nicholas, MBA, PMP

CHARM

Utah Department of Health

September 2002

This publication was supported by a contract from All Kids Count, a program of The Robert Wood Johnson Foundation, to the Utah Department of Health. The views, content and citations reflect those of the Utah Department of Health.

Ordering Information

This publication is available online at the All Kids Count web site, www.allkidscount.org.

Copyright © 2002 by All Kids Count, Center for Innovation in Health Information Systems. All rights reserved.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	V
-------------------------------	----------

EXECUTIVE SUMMARY	VII
--------------------------------	------------

PROJECT DESCRIPTION	VII
APPROACH	VII
CONCLUSION	VII
RECOMMENDATIONS	VII

INTRODUCTION	1
---------------------------	----------

WHY THIS DOCUMENT WAS WRITTEN	1
TARGET AUDIENCE	1
APPROACH	1
HOW TO USE THIS DOCUMENT	1

REQUIREMENTS GATHERING & DOCUMENTATION PROCESS	3
---	----------

REQUIREMENTS GATHERING IN THE CONTEXT OF SYSTEMS DEVELOPMENT	3
MAJOR SYSTEMS DEVELOPMENT PHASES	3
REQUIREMENTS PURPOSE	3
RELATIVE IMPORTANCE OF REQUIREMENTS GATHERING	4
REQUIREMENTS AND SCOPE DEFINITION	5
RESOURCE CONSTRAINTS	5
MODERN DEVELOPMENT PRACTICES	5
SCOPE SELECTION AND JUSTIFICATION	5
SCOPE EXCLUSIONS	6
REQUIREMENT TYPES	6
GRANULARITY/PURPOSE BASED REQUIREMENTS	6
FORMALITY BASED REQUIREMENTS	7
DEGREE-OF-COMPLETION BASED REQUIREMENTS	7
CONTENT BASED REQUIREMENTS	8
REQUIREMENTS GATHERING METHODOLOGY	10
ACTORS OR PARTICIPANTS	10
ACTIVITIES	11
REQUIREMENTS DOCUMENT	14
SOURCING/AUTHORING	14
DATING	15
CHANGE MANAGEMENT	15
VERSIONING	15

PUBLISHING	15
REFERENCING FOR TRACEABILITY	15
<u>THE CHARM PROJECT – A PRACTICAL VIEW.....</u>	17
CHARM SPECIFIC CONSTRAINTS	17
FINANCIAL CONSTRAINTS	17
RESOURCE CONSTRAINTS	17
WINDOW-OF-OPPORTUNITY CONSTRAINTS	17
CHARM REQUIREMENTS AND SCOPE DEFINITION	18
DEVELOPMENT APPROACH.....	18
SCOPE SELECTION AND JUSTIFICATION.....	18
SCOPE EXCLUSIONS	18
CHARM REQUIREMENT TYPES	18
GRANULARITY/PURPOSE BASED REQUIREMENTS.....	19
FORMALITY-BASED REQUIREMENTS.....	19
DEGREE-OF-COMPLETION-BASED REQUIREMENTS	20
CONTENT BASED REQUIREMENTS	20
CHARM REQUIREMENTS GATHERING PROCESS	22
ACTORS OR PARTICIPANTS	22
ACTIVITIES	23
CHARM REQUIREMENTS DOCUMENT	28
SOURCING/AUTHORING.....	28
DATING.....	28
CHANGE MANAGEMENT	28
VERSIONING	29
PUBLISHING	29
REFERENCING FOR TRACEABILITY	29
<u>ACRONYMS</u>	30

Acknowledgements

The author is grateful to the CHARM team for investing of themselves to make the project a success and its ultimate integration goal, holistic care delivery and care coordination, possible.

Special thanks go to:

Stephen Clyde, PhD	Associate Professor of Computer Science, USU
Monal Daxini	Graduate Student, Computer Science, USU
George Delavan, MD	Director, CFHS, UDOH
Jared Dinerstein	Graduate Student, Computer Science, USU
John Eichwald	Program Manager, CHARM
Don Gabriele	Prior Project Manager, CHARM
SubAsh Gutala	Graduate Student, Computer Science, USU
Margaret Lubke, PhD	Senior Research Scientist, NCHAM
Barry Nangle, PhD	Director, Center for Health Data, UDOH
Aaron Rice	Graduate Student, Computer Science, USU
Sharon Talboys	Information Analyst, CFHS, UDOH
Aihua Tong	Programmer Analyst, CHARM
Arvind Varma	Graduate Student, Computer Science, USU
Mike Webb	Project Manager, CHARM
Karl White, PhD	Professor of Psychology, USU; Director, NCHAM

(in alphabetical order)

Additional thanks go to all the Program Managers and staff who are working on the integration effort and who have contributed their time, effort, knowledge and faith to the success of this project.

Executive Summary

Project Description

This study is based on the particular experience of the Child Health Advanced Records Management (CHARM) project, which is a system integration initiative of the Utah Department of Health (UDOH). The project is using an iterative & incremental approach to systems development where refinements continue to be applied to the artifacts produced in earlier iterations. The proposed architecture uses middleware to manage online, real time communication and data sharing among a number of systems already in place. The initial release of CHARM will integrate Electronic Birth Registration, Newborn Blood Screening, Newborn Hearing Screening, Immunization Registry and Early Intervention.

Approach

The white paper starts by considering the theoretical aspects of the requirements gathering process in order to lay the foundation for the case study to follow. In the early sections, requirements gathering activities are placed in the context of systems development and their artifacts are discussed in terms of types and significance.

The case study presented in the later sections juxtaposes this formal approach with what often happens in real life situations. The study has been developed enough for the reader to appreciate the differences that can occur between theory and practice and it has been based on the actual experience of the CHARM project.

Conclusion

All projects have to operate within a number of constraints. CHARM is no exception. Due to resource limitations, primarily financial in nature, the CHARM project team has had to make deliberate and early choices whether it will be caught in a game of perfectionism or accept to move forward when adequacy has been achieved. Adequacy has been chosen in most cases, except in critical artifacts, such as architecture design or technology choices, where excellence was a mandate. In addition, because of its iterative & incremental approach, CHARM has been in a position to continue to refine its artifacts and to approach perfection through ongoing refinements.

Recommendations

While we hope that this experience can be put to good use by other organizations starting out in data integration initiatives, we do not purport that the UDOH practical solutions are the best or even recommended. Specific conditions and local constraints often dictate the detail, formality and even order in which certain activities are performed. However, the UDOH and the members of the CHARM project think that others can learn from these experiences even though they may operate under different conditions. It is in this spirit that this paper was written.

Introduction

Why This Document Was Written

This case study and the related white paper were funded by the Center for Innovation and developed under the aegis of “Connections”, a collaborative among several public health organizations, involved in data integration initiatives. This study uses the particular experience of one of these organizations, the Utah Department of Health (UDOH) and its Child Health Advanced Records Management (CHARM) data integration initiative.

Based on their specific circumstances, projects have to decide how closely they can follow theoretical systems development precepts versus more practical approaches. This case study does not propose that the CHARM approach be taken as best practice. What we offer as a lesson learned is that all projects are different and that local conditions and constraints often dictate a less formal or a less rigorous approach be pursued. That may make the difference between reaching the end of the road or getting bogged down along the way.

Target Audience

As state and federal funding is becoming available for projects that pursue data integration in public health, many public health organizations are now planning to engage in such integration efforts. Some are better staffed than others with resources that are knowledgeable in informatics or in system development. UDOH is going ahead with its CHARM project without any state funding under somewhat severe financial constraints. This white paper may be more applicable to project teams that are eager to start out their integration projects but are staffed at less than ideal levels of skill and experience and, because of that, are likely to be less process-oriented and more results-oriented in their approach.

Approach

The white paper will first cover the theoretical content needed to lay the foundation for what the requirements gathering process is, to place it in the context of systems development, and to discuss the types of requirements and their significance. It does not purport to rise to the level of a manual on requirements gathering. It will be developed just enough for the reader to appreciate the differences that can occur between theory and practice.

The case study sections will show how Utah applied the theory in practice, within a concrete context. It looks at how theory meets practice in real life and some of the reasons why the two are often different.

How To Use This Document

What the experience of the UDOH shows is that practice is often different from theory. What we recommend the audience learn from this is not that the theory is

wrong and that is why one has to do something else in real life, but that the theory does not always consider the constraints of real life. What the theory calls for, through its rigor and completeness, can be called “the rich man’s” version. In real life, one never has enough resources to do things “by the book” and one often has to cut the process down to size. We will call this “the poor man’s” version. What we expect the readers to figure out is where they might be well served to take shortcuts and where they might get into trouble.

Requirements Gathering & Documentation Process

Requirements Gathering in the Context of Systems Development

This section discusses the larger context within which requirements gathering is performed. Although “requirements gathering” is the typical terminology used in the context of systems development, it is often called “requirements definition” and is not dissimilar from what is known as “needs assessment”. They all deal with the identification and documentation of mismatches between what people have and what people need and they are the basis on which organizational growth and change is predicated.

Since most data integration projects are technical projects dealing with systems development activities, one can look at requirements gathering as a major set of activities executed close to the beginning of the systems development project.

Major Systems Development Phases

Following is a fairly traditional sequence of the major activities involved in systems development. They establish the context within which Requirements Gathering should be considered.

Pre-Project Activities

1. Problem Definition
2. High Level **Requirements Gathering**
3. Business Case Development / Solution Definition and Feasibility Study
4. Proposal Review, Approval and Funding

Project Activities

5. Requirements Definition (includes Detail **Requirements Gathering & Analysis**)
6. Design (includes Architecture, Data and Functional Design)
7. Component Development (includes coding and unit testing)
8. System Testing (Alpha Testing)
9. User Acceptance Testing (Beta Testing)
10. Deployment & Maintenance

As many will argue right and left of this sequence with respect to completeness, accuracy, order, etc., its purpose here is to illustrate where in the process Requirements Gathering activities fall – in this case, steps 2 and 5.

Requirements Purpose

Requirements are of paramount importance in the systems development process. They capture, first hand, the unmet needs of the organization commissioning the effort. Requirements gathered at a high level constitute the basis for the development of a justification, or a business case, to secure project funding. Requirements gathered at the detail level document the users/stakeholders’ specifications for what the system will need to provide.

Since requirements will become the basis on which all the costly resources will be engaged and all the systems development work will be conducted, all gathered requirements will have to be validated and prioritized by the users. Once documented and validated, requirements will be organized by type, analyzed and documented. As such, they will be used as the basis on which the system will be designed and, ultimately, tested. The degree to which the implemented system meets the users' requirements constitutes the acid test for a system's completion and user acceptance.

Relative Importance of Requirements Gathering

As already identified under Requirements Purpose, the users' requirements describe, in business terms, what the system is expected to provide in order to resolve the problem, as defined.

Statistics show that systems development projects fail at a fairly high rate, with disastrous consequences in wasted resources, compromised reputations, and unsolved business problems. One of the most frequent causes relates back to requirements. Since so much in systems development is directly related to requirements, failure to gather requirements correctly and completely and failure to validate them often leads to project failure.

Requirements also have a tendency to evolve. Business conditions change, new regulations are being enforced – the “problem” evolves. Because systems development is a fairly lengthy process, by the end of the project, requirements gathered at the beginning of the project no longer seem to meet the users needs. This has often been the case in more traditional systems development where contact with the users is limited: users are involved in requirements gathering early on and then they re-emerge at the end of the project, in order to perform user acceptance testing.

In newer approaches, such as Rapid Application Development (RAD), users are kept closer in the loop. In addition to providing the initial set of requirements, they are also involved in the design and development of the prototype where they have ongoing opportunities to shape the system. This insures requirements are captured and modified along the way. While this is desirable from a system fit perspective, it creates stress in project management and project change control. Additional requirements often translate into project scope creep, in other words, scope expansion, within the context of limited time and money. In order to accommodate the new scope, that is the new requirements, projects often take longer and cost more than initially estimated.

Modern approaches, such as Iterative & Incremental Development, assume that the phases cannot be completed sequentially. Instead, projects are completed in a series of cycles. Each cycle, called an iteration, goes through the all the project phases but places less and less emphasis on early phases and more and more on later phases. This assures that evolving requirements continue to be uncovered during future iterations.

Requirements and Scope Definition

Resource Constraints

All projects, indeed, all human endeavors, are constrained by the limitations of available resources, usually expressed as time and money. Because of that, the scope of human endeavors is, by necessity, limited to what can be achieved within a certain timeframe and within certain budgets. Projects are no different.

Once the **scope** of what can be achieved has been identified within an associated **timeframe** and **budget**, these three begin a tug-of-war. With respect to systems development, scope is best expressed as the totality of the requirements to be included in the solution. Even with all the attention given to requirements ranking and initial scope definition where highest priority requirements are included in early releases and lower priorities are included in later releases, scope will still tend to change through time. As we all know, today's requirements will be different tomorrow and the users/sponsors will want to include these new/changed requirements in the solution, or else the solution will not meet their new needs. The project leader/manager has the difficult task of managing this changing scope such that commitments made with respect to time and budget are being kept. Naturally, this is very difficult and there is a constant tug-of-war between these three.

Modern Development Practices

Problems with such “new” requirements, usually called “scope creep” are most often associated with waterfall development approaches. As discussed in the section on Degree-of-Completion Based Requirement Types, waterfall methodologies collect requirements only once, at the beginning of the project. During all the subsequent stages of the project, the project leader/manager has to wrestle with the users/sponsors and keep them from making additions and/or changes to their requirements. As project leaders/managers often cannot but give in to these pressures, the scope grows and the projects end up exceeding their time and budget commitments.

In modern iterative approaches, scope growth is taken for granted and is being handled through quick successive design and development iterations and through rapid system releases. Beyond an initial core of requirements that the users identified coming in, new requirements are continuously gathered and refined, as users are involved throughout the process, in design and prototype development. Each new release has its own schedule and budget expectations and as the users want to get more functionality, they are usually prepared to provide additional resources (time and money).

Scope Selection and Justification

Outside of inherent dependencies between certain requirements, the users/sponsors have the freedom to dictate what requirements will be included

and what their relative priority should be in the system release sequence. This is done, for the most part, during the activities described in the section on Prioritization Activities.

If a business case and a feasibility study had been completed prior to requesting the project's approval and funding, an initial set of rough, high level requirements had been included and justified, as part of those activities. Otherwise, this would have been done as part of the Project Chartering activities.

Scope Exclusions

Areas that are easily assumed to be normal extension of the scope and are included in the scope should be so identified. On the flip side, areas that are easily assumed to be normal extension of the scope but are not included, should be so identified as well. These will be documented under a section normally referred to as "Scope Exclusions".

Requirement Types

Granularity/Purpose Based Requirements

High Level Requirements

High level requirements are, typically, the initial set of requirements. They are usually formulated in broad strokes and are sourced from executive sponsors and stakeholders. They are primarily used for feasibility studies. Feasibility studies do not aim at building the solution and, therefore, do not need detail requirements. In addition, feasibility studies are targeting the executive level and must stay at a fairly high level. Feasibility studies do, however, need requirements in order to frame the problem and the solution, to provide a few implementation options and to ask for approval, prioritization and funding for recommended option.

Detail Requirements

Detail level requirements are further specification of the high level requirements and constitute the complete formulations of the solution needs. They are sourced from the actual Subject Matter Experts that have intimate knowledge of the problem and of what the solution needs to address. Detail requirements are used for systems development activities, such as solution design, specification and testing. Because errors in detail requirements affect the delivered system, detail requirements need to be documented and verified with the sourcing Subject Matter Experts to ensure accuracy and completeness.

Formality Based Requirements

Based on the degree of formality in their gathering, formulation, documentation and validation, requirements can run the gamut from formal to informal. While, at one end of the spectrum, there are “formal” requirements and, at the other, there are “informal” requirements, depending on project circumstances, actual requirements fall anywhere along that spectrum.

Formal

Formal requirements are the result of a rigorous, disciplined, pre-determined process. Requirements documents are published and appropriately versioned. Sourcing information as to who authored each specific requirement is tracked for later traceability back to source. Formal requirements are very important in large organizations with big IT shops where requirements gathering and documentation is the responsibility of one group, usually reporting to the project sponsor, while system development is the responsibility of another group, usually the IT Department. The requirements document is what is actually handed off to the development group. Formal requirements are also very important when system development is contracted out to a vendor.

Informal

One situation when requirements qualify as “informal” is when they are communicated verbally and through note taking. Written requirements may also qualify as informal when they do not quite have all I’s dotted and T’s crossed and when they follow a much less rigorous and protocol-laden process. Informal requirements are the norm in small shops where there is little available staff and there less opportunity for role/functional specialization – staff does it all. As a consequence, there is also little need to hand-off the requirements document to another party or group.

Degree-of-Completion Based Requirements

The systems development methodology followed by the project can also influence how requirements are gathered and what the end product looks like. Actually, it is less a matter of what type of requirements are being produced than of what their degree of completeness and finality is.

Final

In traditional systems development approaches, usually called “waterfall” approaches, each systems development phase has one beginning and one end. No phase in the process sequence can start until the one before it has completed. In such approaches, each activity has to make sure that what it produces and passes on to the next phase is as final and as complete as possible. Once the hand-off takes place, the previous phase activities and outputs are “frozen” and not subject to change.

The “frozen” and final requirements may be convenient in terms of reference, but they usually generate problems down the road when the system is being delivered to the users. In most cases, requirements frozen, say a year before, no longer solve the dynamic work issues of today.

Evolving

Newer systems development approaches, such as iterative & incremental development (see section on Modern Development Practices), follow repeating (iterative) cycles through the entire process while developing each area more and more (incremental). In such approaches, during the first iterations, more effort is spent on the early phases of the process, which formulate the solution. Then, during later iterations, more effort is spent on the later phases of the process, which build the solution.

By going through an entire cycle, iterative & incremental approaches discover additional facts and the next iteration is likely to change, enhance, “increment” the products of the previous iteration. That makes the requirements, as well as any other product, “evolving”. While evolving requirements are fluid and therefore more difficult to manage and document, they are more likely, in the end, to meet the users’ needs in a dynamic business environment.

Content Based Requirements

All requirements need to cover the content of what the users need. Within the variety of this content, there are usually a number of topics around which most of the content-based requirements naturally organize. The organization by topics is helpful because it groups requirements around focus areas thus facilitating review, analysis and design activities. While this organization is subject to a lot of variety, below is a simple yet common organization by type of content.

Data Requirements

Data requirements identify, describe and organize:

- what data entities are of consequence to the application;
- what data elements are to be used, changed, created or persistently stored (attribute analysis);
- what are the sources of data elements;
- what business function uses each data element.

These requirements are documented and further analyzed and organized in data (or object) models, using various data analysis and modeling methods and tools.

Functional Requirements

Functional requirements capture what needs to be done with the collected and stored data. The business logic and workflow identified by the users are documented and further analyzed and “scripted” in function models, and even in storyboards, following various methodologies and using various analysis and modeling tools. Object oriented methodologies capture these requirements in “use cases”. While specifying the actions being performed, the function models, or use cases, usually identify and reference the data items used and the organizational roles playing a part in executing each function.

Constraints

This is a “catch-all” category of requirements. Since it deals with a variety of topics, this is where most of the variety in organization is derived from. Some projects choose to break this catch-all category and set up a number of smaller topics instead. Constraint-type requirements comprise:

- access requirements such as:
 - location access requirements;
 - user access requirements (who the users are, what access channel they can use, what functions they can perform, what data they can see, etc.);
- legal constraints, such as statutes and/or rules that affect the system;
- usability requirements:
 - type of user interface used on screens, reports, AVR;
 - ADA requirements;
 - online tutorials and/or how-to wizards requirements;
- capacity requirements, such as:
 - data volumes;
 - transaction volumes;
- requirements for integration or interoperability, such as:
 - interfaces;
 - messaging;
 - communications;
 - use of common/shared identifiers;
 - shared (public) vs. local (private) data;
 - services the new solution will provide to the enterprise;
 - enterprise services the new solution will use;
- data migration requirements, such as:
 - data source;
 - data content (elements)
 - data conversion and/or cleansing needs;
 - migration sequence;
- security requirements, such as:
 - firewall protection;
 - levels of user authentication;

- levels/types of encryption needed;
- audit trail and level of detail;
- standards that need to be adhered to, such as:
 - data standards;
 - architectural standards;
 - network/communication standards;
 - tools and technology platform standards;
 - HW/SW constraints (e.g. specific HW/SW needs to be used);
 - any necessary deviation from enterprise standards and justification;
- operational requirements such as:
 - system availability requirements (e.g. 24x7, 8-5/Mo-Fr, etc.);
 - support availability requirements (may or may not mirror availability requirements);
 - frequency and type of backup (i.e. how often and how complete);
 - “critical” system classification (dictates the pecking order for recovering system after failure);
 - response time expectations;
- product and component delivery schedules and deadlines.

Requirements Gathering Methodology

The methodology of gathering requirements describes the process that is being followed and the main process elements that need to be considered:

- who the actors are;
- what activity is being performed;
- what goes in the activity, or the input;
- what comes out of the activity, or the output;
- when the activity is being performed;
- what the significance of the activity and its output is.

The actors will be discussed in a separate section on Actors or Participants. The other process elements will be looked at together, as each activity is discussed.

Actors or Participants

Business/Systems Analyst

Requirements gathering is, typically, the specialty of the business analyst, or of the business/systems analyst, depending on what title has been adopted. This resource is usually part of the sponsors’ (or the users’) organization and is knowledgeable about the business, the problem and the solution the users are looking for. The business /systems analyst also has a comprehensive understanding of all aspect of the systems development process and acts as an interpreter between the users and the IT developers. Their role is to protect the users’ interests, to diligently gather, document and validate the

requirements and to ensure their faithful translation and understanding by the IT development team.

Facilitator

When large systems are developed, gathering requirements is usually performed using specialized techniques, such as Joint Application Design (JAD) or focus groups. In either case, at least one facilitator is needed to help the group stay on track, work well as a group, and achieve its goals. Facilitators are trained in the use of facilitation techniques.

Project Leader / Project Manager

Not only is the project leader/manager involved in the planning, scheduling and preparation activities, but they are the primary liaison between the development team, on one hand, and the sponsors, users and stakeholders, on the other. Since requirements gathering is a stage in which the users are most heavily involved, it is easy to see why the project leader/manager would have to play such a prominent role. In addition, since they have the responsibility for moving the project forward while ensuring quality and completeness at every stage, they are heavily involved in the activities that prioritize, validate and sign off on the requirements.

Subject Matter Experts

The users are also active in the requirements gathering process. They are usually represented by a number of seasoned users, called Subject Matter Experts or SMEs. The SMEs need to be experts at the business, be fairly articulate, and be allowed to spend the necessary time with the business analyst(s) and other project team members, as needed.

IT Development Team

The IT team should be available and provide reality checks to the business/systems analysts in terms of the reasonableness and feasibility of some of the more unusual requirements, such as technical requirements or constraints.

Activities

Activities involved in requirements gathering include planning and preparation activities, actual requirements discovery and collection activities, prioritization activities, and validation activities. In many methodologies, requirements analysis is often included either with requirements gathering or immediately after. This document will not include requirements analysis activities.

Planning and Preparation Activities

These activities precede the actual requirements collection activities and involve:

- Determining what tools and techniques will be used and who will need to participate;
- Preparing instruments to be used, e.g. questionnaires, interview scripts, checklist, etc.
- Scheduling requirements gathering activities;
- Preparing and lining up resources;

These activities are performed by the Business/Systems Analyst. The Project Leader/Manager will ensure they all fit within the allocated time and effort on the project schedule. Facilitators will need to be selected and involved if any group techniques are chosen. The SMEs will also be involved in preparation activities.

Based on special inputs, such as 1) the project charter, 2) problem statement and 3) project scope, a number of outputs will be generated during this activity, such as:

- list of selected tools and techniques,
- list of participants,
- schedule of activities,
- interview scripts,
- checklists,
- templates,
- focus group or JAD agenda and instruments,
- material for the training and preparation of resources, such as the SMEs.

Completing these activities will ensure that the requirements collection and documentation to follow will use the most suitable tools and techniques, will have appropriate and customized instruments and will have all resources on notice, prepared as to what is expected and lined up. If the users or the SMEs involved are new to the techniques of requirements gathering, preparatory one-on-one or group sessions will need to be conducted so people are productive and know what to expect.

Requirements Discovery & Collection Activities

These activities represent the core of the requirements gathering stage. They are conducted by the business/systems analyst and by the facilitator and they involve the identified SMEs.

These activities use the tools, instruments and techniques previously selected and prepared and they include:

- Probing in each knowledge area with the help of knowledgeable SMEs;
- Conducting interviews, JADs, focus groups, or using other formats to discover what is needed to solve the problem as identified;

- Capturing and organizing all the collected information;
- Verifying accuracy with the requirements' authors.

The outputs of these activities are the actual requirements, although, at this point, they will look very raw and disorganized. Authoring information will be attached to each requirement.

Having the requirements identified and labeled with authoring information constitute the basis for formulating the design and ultimately implementing a solution that has been "specified" by the users. Authoring information is also used for verification that requirements have been truthfully recorded. This needs to be done with quick turn around, before short-term memory fails to recall.

When conducting group activities, such as JADs or focus groups, additional benefits are often reaped when the project team gels together as a unit. That is good for the project.

Prioritization Activities

Once the collected requirements have been captured, documented and validated, they often add up to more than can be included in the solution, at least, with the initial time and budget. Users are faced with the very heart-wrenching task of choosing between needs. Various sorting and weighting techniques are used and the most urgent requirements usually emerge to the top of the list.

Since requirements are often dependent on one another, requirement dependencies will have to be identified prior to any prioritization effort.

The business/systems analyst normally performs this activity. Then the SMEs are asked to prioritize. However, if prioritization looks like it can be a politically-laden exercise, it is best performed in a group setting, most likely a facilitated session, and a facilitator will also be involved.

The individual SMEs, or the group, will use, as input, the requirements as gathered and grouped around interdependencies and will score, weight and prioritize each requirement. The output will be the same document where each requirement now has a prioritization weight and score.

This document is very significant since it not only provides what the users need but it also allows the solution to be phased in according to the users' most pressing needs. The project leader/manager can use this information to estimate the effort required and identify what the project's initial scope can be and what later releases will include.

Validation Activities

Before the requirement can officially become the basis for further design and development work, they need to be formally validated and accepted by the

users. This is accomplished through reviews and walkthroughs with the SMEs.

Several sub-activities are included:

- Producing the requirements document;
- SME's private review of the document;
- Conducting a Requirements Walkthrough;
- Revising the document as needed;
- Obtaining SMEs and sponsors' signoff.

Before the actual validation can occur, the business/systems analyst will use the requirements, collected, verified for accuracy, grouped by dependencies, and prioritized, to create a complete requirements document. The document usually follows the functional organization described in the section on Content Based Requirement Types.

The business/systems analyst will schedule a validation session with the SMEs and the project sponsors and will send the requirements document out for review prior to the session. The project leader/manager usually attends this activity and participates as needed. During the validation session, the business/systems analyst will conduct the requirements walk-through interpreting each one so that the SMEs can provide validation or correction, as needed. After the session, the business/system analyst will make appropriate revisions to the document and will collect signoffs from the SMEs and from the project sponsors.

This is the official "OK" for the project to use these requirements as the basis for its design and implementation activities.

Requirements Document

Activities associated with documenting the requirements occur, in various degrees of "roughness", all through the process of discovering, collecting, grouping, and ranking the requirements. Formal documentation is completed in preparation for requirements validation, as identified in the section on Validation Activities. Once formalized, validated and signed-off, the Requirements Document becomes an official document of the project.

A more utilitarian version of the requirements document is often used to maintain various pieces of information as identified below. This document is often used when clarifications need to be obtained from the resources that authored the requirements.

Sourcing/Authoring

It is critical that information on who made, that is who authored, a given requirement be retained. Often, requirements are being challenged by other users/sponsors, especially when requirements conflict or when turf battles are fought for what is included in scope. If information on the author has not been retained, it is difficult for the business/systems analyst to justify requirements.

Authoring information is also needed for further clarifications. It is often impractical for the business/systems analyst to capture all the necessary detail associated with a requirement. Ability to access the source later on is critical.

Dating

Because of the same need for further clarification, validation and justification of the requirements, the documentation should also include information on the circumstances in which requirements were gathered, such as date/time and even place or event, e.g. JAD, focus group, individual interview, etc. Dates also help place requirements in the proper time perspective, especially as requirements change.

Change Management

Like any other official project document, the requirements document should have a change tracking section, usually placed close to the beginning of the document. That is where versions and major changes to the document are identified and tracked – this section of the document is called “Change Management”. The Change Management section tracks information on who made changes to the requirements document, what was changed, and the date of the change.

Versioning

Changes cause the document to evolve. With each set of changes, a new version of the document is being generated. Versioning is used in order to avoid confusion by indicating which set of requirements is being referenced. Requirement changes have to go through a similar validation process as the original set of requirements did.

Publishing

Publishing is the final step in making the requirements known. Depending on the project visibility and how the implementation effort will be carried out, “public” may range from the immediate project team, such as SMEs, users and sponsors, to vendors and even the public. The actual format will depend on the audience and will range from MS Word documents in loose-leaf binders to professionally published and bound documents. Formal documents are hard to maintain, especially in cases of highly volatile requirements and frequent versioning.

Referencing for Traceability

Based on what has been presented so far, it is clear that the requirements document needs to contain certain important pieces of information and needs to be organized in a way that is easily followed and comprehended. In addition, a good numbering and reference scheme should be used such that each requirement can be easily referenced.

Requirements are being gathered and documented with the specific purpose of ensuring that the system to be developed actually meet each one of these needs. During the design process, traceability checks are being performed in order to ensure each requirement included in scope has been properly addressed. The first part of the traceability process is to note the reference number of each requirement addressed in the design. The second part is to verify that all requirements have been traced back.

Traceability is also serving during the testing phase. The point here is to ensure that each requirement has been included in the system and that it behaves as expected.

The CHARM Project – A Practical View

CHARM is an acronym for Child Health Advanced Records Management. It is a project of the Utah Department of Health (UDOH) which integrates several standalone child health systems. It has been selected as the case study for this white paper. Since CHARM is somewhat typical of projects that operate under real life constraints yet it has been fairly successful up to this point, it can offer a practical perspective to be juxtaposed to the theoretical view. CHARM has gone beyond the initial requirements gathering phase and is currently completing its third architecture proof-of-concept iteration.

CHARM Specific Constraints

From the beginning, CHARM has experienced several constraints that have slowed its progress and have limited the amount of time the team could spend on interim deliverables. In spite of this, CHARM has been able to overcome these limitations by “cutting corners” and doing only the minimum necessary to keep the project moving forward.

Financial Constraints

There is no arguing that financial constraints are part of the limitations most projects have to deal with. CHARM started with no budget and, for almost a year, it moved along only through the sheer passion and dedication of UDOH employees who had full time jobs to manage at the same time. Starting in June 2000, UDOH was awarded its first grant that had a data integration component. A few other grants awarded since then have allowed the CHARM project to engage a few part time IT resources.

Resource Constraints

Resource constraints are the direct result of financial constraints. Because of limited financial resources, the project could not afford trained and qualified resources such as business/systems analysts to engage in the requirements gathering activities. Also, because of consecutive budget cuts over the past few years, program resources have been cut to subsistence levels. That drastically limited the time availability of SMEs.

Window-of-Opportunity Constraints

Window-of-opportunity constraints are a special type of time constraints. They are more relevant to projects that are moving more slowly because of financial and resource constraints. Window-of opportunity constraints dictate that a certain tempo be maintained such that the project does not lose momentum and stakeholders do not lose interest. CHARM has experienced this type of constraint. It calls for the project to move forward even though previous deliverables have not been completely polished.

CHARM Requirements and Scope Definition

Development Approach

With its modern, iterative & incremental approach, CHARM expects, from the beginning, that the requirements gathered initially will evolve and will grow. It is also with the intent of speed and efficiency that requirements are discovered and rendered in quick short cycles. With a flexible, scalable and component-based architecture, it is relatively easy to add functionality in response to “new” requirements. It is also the intent that additional requirements will be prompted by the actual use of the system. CHARM proposed a “low bar” entry to the programs, with the expectation that the bar can be raised by the programs issuing additional integration requirements.

Scope Selection and Justification

CHARM’s “low bar” entry is actually a reflection of the programs guarded acceptance and adoption of the integration concepts and capabilities. While CHARM has bold plans for its future releases, it is going about it in baby steps.

Its scope selection has two dimensions: one looks at who will be a participating entity, the other looks at what data and what functionality will be supported. As part of the requirements gathering, the entities targeted for participation were evaluated according to a number of readiness criteria and were grouped in clusters. The initial release includes the cluster deemed most “integration-ready”: Vital Records (Births and Infant Deaths), Newborn Blood Screening, Newborn Hearing Screening, the Utah Immunization Registry (USIIS), and Early Intervention.

Also as part of the requirements gathering, these initial five programs were the basis for determining what data and functionality will be needed. All the requirements gathered during the first iteration can be covered in the scope of the first release. In general terms, all the high level requirements of CHARM are “in scope”.

Scope Exclusions

Given the flexible architecture on which CHARM is being developed, nothing has yet been identified as out-of-scope. The other programs targeted for inclusion in CHARM but identified for later releases are in-scope.

CHARM Requirement Types

Below is a discussion on what types of requirements the CHARM project has produced and why.

Granularity/Purpose Based Requirements

High Level Requirements

In order to provide service in a more customer-centric manner, the UDOH held a focus group retreat with management and executive level participation. The focus group identified five high level, high visibility integrative initiatives and assigned sponsors/owners for each. One of these initiatives was the “Early Child Preventive System Integration”. This is how the CHARM project got started. During the focus group, the initial set of critical success factors was developed. The owners/sponsors further developed these critical success factors. They became the CHARM high level requirements and were published using the format of the National Immunization Program (NIP) minimum requirements for immunization registries.

Once the CHARM project was able to put a small project team together, these high level requirements were used to conceptualize the architectural solution for the CHARM integration.

Detail Requirements

The CHARM project developed and published an initial set of detailed requirements. They were sourced from the program staff who were best equipped to identify what data sharing needs they had and what they would be able to do with the shared data. Legal counsel also participated in ensuring the legality of sharing various data among programs.

The CHARM detail requirements have been used in developing the CHARM design and will be used again, during user acceptance activities.

Formality-Based Requirements

The CHARM project followed a mix of formal and informal requirements gathering activities. With funding from a Needs Assessment grant from HRSA, several formal facilitated focus groups were held to gather user and stakeholder requirements. The ensuing documentation also followed formal presentation guidelines as required by the grant. To supplement the requirements gathered through the focus groups, a series of informal interviews were conducted with the program staff.

Overall, because of resource constraints, CHARM requirements followed a fairly informal process during gathering, formulation, documentation and validation. With the diminutive size of the CHARM project team, it would be quite impossible to expect formality in much that they do. One pays a price for formality and, when resources are so limited, formality can only come at the expense of content.

While both the high level and the detailed requirements have been gathered, published and reviewed with the users, this has been done with a minimum

expense of time and effort. Outside of the initial facilitated focus groups, the team did not have a business/systems analyst to drive the requirements gathering activity. The role was played by an Information Analyst and by a counterpart resource from USU, UDOH partner on the CHARM project. A lot of information has been gathered but much is still in the form of notes

As will be seen in the section on Degree of Completion, accepting less formal requirements is compensated by the fact that the requirements are allowed to evolve, through iterative & incremental development. In addition, CHARM does not involve an outside vendor. It has a rather small team and communication flows with relative ease.

Degree-of-Completion-Based Requirements

Because CHARM is following an iterative & incremental development approach, its requirements continue to evolve during each iteration. That means that the next iteration set and version of the requirements is likely to be an evolution of the previous set and version. This is one of the main reasons why a less formal approach to requirements is acceptable.

While requirements are prone to evolve due to the iterative aspect of the development approach adopted by CHARM, additional requirements evolution is expected. It is anticipated, and actually quite likely, that the programs targeted for initial participation in the integrated solution will, in time, develop an enhanced understanding and appreciation of the potential and value of the integrated solution. This will create a new generation of requirements that would have been inconceivable before.

Content Based Requirements

The organization of the CHARM requirements follows roughly the organization identified in the section on Content Based Requirement Types.

Data Sharing Requirements

Since CHARM is actually a data integration project, the Data Requirements section has been called “Data Sharing Requirements” and it actually focuses on the data sharing needs of the participating users, both in terms of what data and in terms of what will be done with the shared data. These requirements identify:

- what data will be included in the shared Child Health Profile (CHP);
- where the CHP information will be stored;
- how the CHP will be initially populated from existing data;
- how CHARM will identify a child;
- what data each program will be able to access and how this should be tracked;
- which data will have precedence for accuracy and reliability;

- how the system will keep track of updates in the participating programs;
- what common services should be provided to all participants;
- what services each participant should be able to access and to provide;
- what data should be included in matching and de-duplication procedures
- how merging and de-duplication should be performed;
- what data cleansing and normalization services should be provided
- what conditions should trigger and remove alerts and reminders.

Functional Requirements

Because of the heavy data sharing focus of the CHARM project, functional requirements were not organized as a separate topic. The functional requirements that were identified were grouped around and included under the “Data Sharing Requirements”.

Constraints

Because of the industry’s concerns about privacy and confidentiality, a section on “Privacy, Confidentiality and Security Requirements” has been included in the CHARM requirements. They cover many of the typical requirements identified in the section on Constraints, under:

- access requirements;
- legal constraints;
- security requirements.

It also includes additional requirements related to:

- informed consent and
- opt-out capabilities.

Another section is entitled “Administration Requirements” and includes requirements under such topics as what system household activities will be available and who will carry them out.

Another section, entitled “Design-Process Requirements” identify constraints imposed on the design process and the methodology used, in order to ensure system usability, extensibility and scalability.

A section on “Development and Testing Environment” identifies “good-practice” constraints on setting up the various system environments and on how testing should be conducted.

A last section identifies “Operational Requirements” that would need to be available for the CHARM system.

CHARM Requirements Gathering Process

Actors or Participants

Business/Systems Analyst

The CHARM team did not have a traditionally trained business analyst, or business/systems analyst to conduct its requirements gathering activities. The project was lucky to have a Public Health information analyst (IA) join the team, on part time basis. While the skill sets were not identical, there was good transferability. In addition, as a self-starter, the IA was soon able to take charge. As a member of the Data Resource Group in the Division of Community and Family Health Services, she brought general knowledge of data and of CSHCN programs. Her assignment to CHARM fitted well with her regular job description. Guidance was provided, as needed, by the project team members.

The IA worked in tandem with another part-time resource from USU, primarily an analyst/researcher. The USU resource brought a background in early intervention, data systems, and web development. Aside from public health backgrounds, both resources had social science backgrounds and good interviewing skills, which were valuable during meetings with the program managers, who represented the project SMEs.

The initial program interviews took place over a 3-4 month period. During this time, interviews were documented, information gathered was processed and made available online for the entire project team. These activities took roughly 30% of the analyst's time.

Facilitator

Facilitators were a great resource for the requirements gathering process. The CHARM project was lucky to have access to facilitated settings twice: early on, when the CHARM high level requirements got their roots and a second time, when focus groups were conducted and most of the detail requirements related to data, privacy and confidentiality were gathered.

Project Leader / Project Manager

The CHARM had a project manager who was assigned to the project only on a part time basis. Because the CHARM project leader also had business analyst experience, he was involved in the requirements gathering activities and provided support and assistance to the IAs. He also helped identify and formulate most of the more technical requirements associated with the CHARM system. He also organized the activities related to the requirements documentation and validation.

Subject Matter Experts

The CHARM project team did have access to user representatives and SMEs but access was very limited in terms of time availability. Successive budget cuts had increased the workload of the program staff to the extent that it was close to impossible to get time with them. Most of the time that was used in drawing requirements was practically stolen.

With advancements in the project's design, the SMEs are providing a little more time to the project and this will be invaluable as the project goes through additional iterations.

IT Development Team

The CHARM IT team is made up of several part time resources who are making the project their personal priority. The team is distributed. Two members are employed by, and located at the UDOH in Salt Lake City but are allocated to more than one project. Four more members are comprised of three graduate students and their professor from the Computer Science Department of the Utah State University in Logan, two hours north of Salt Lake City.

Activities

All the types of activities that comprise traditional requirements gathering were performed by the CHARM project team and all the expected deliverables, both interim and final, were created. The difference is that the activities were conducted with a lot less formality and the deliverables would most likely lose points on presentation.

Planning and Preparation Activities

Planning and preparation activities for the IAs included background research on the concepts of integrating client data. This activity assisted in the documentation of potential benefits and outcomes of a CHARM-like system. Although few examples of positive benefits or outcomes were found in the literature in the health sector, the literature on the assessment of total quality management (TQM) provided a framework in which to evaluate the CHARM application, so that benefits and outcomes can be measured once CHARM is implemented.

Concurrently with the background research activities, data modeling training was provided to many of the UDOH staff. Upon completion of the course, the UDOH IA was able to use the basic features of a data modeling software tool.

In anticipation of the actual requirements gathering activities, the two analysts prepared interview guides and met with the program staff to outline the upcoming activities and their major objectives. This provided the program

managers an opportunity to better understand the scope of the project and to plan for providing the necessary information.

Requirements Discovery & Collection Activities

Prior to the detail requirements gathering, the UDOH had conducted a preliminary needs assessment effort with representatives of the user groups CHARM was expected to address. In the first release, these groups represent primarily internal UDOH programs. The assessment was carried out using focus groups planned and conducted by a trained facilitator. The sessions focused on data sharing. A great number of data sharing opportunities were discovered and documented, in terms of what data, from what program, what could be done with that data which would otherwise not be possible. This represented an initial core of data sharing needs.

When the actual project team was pulled together, a number of other activities further gathered and refined these requirements.

The actual requirements discovery and collection activities for the CHARM project included several activities and were conducted in several stages.

The main activities included:

1. Confirmation of the CHARM integration concepts with the programs;
2. Identification of the programs' business needs;
3. Gathering data and developing data dictionaries;
4. Identification of the programs' data sharing needs and limitations;
5. Clarification of data rules, authority, and default options;
6. Development of data models for the base data and for the shareable data;
7. Development of data sharing agreements;
8. Identification of each program's service offering and service needs.

The detail activities included:

1. Confirmation of the CHARM integration concepts with the programs

It took some time, initially, to communicate the purpose of the CHARM project and its planned capabilities. While the managers of the programs involved were already familiar with the CHARM vision and high level requirements, integration was still a new concept. It was very important to take this time to build relationships, gain an understanding of the organizational structure and culture, and strengthen the programs' buy-in into the CHARM project.

2. Identification of the programs' business needs

Two-to-three interviews were conducted for each program in order to understand the roles and the needs of each program and to identify where and how the participants fit into the project. The interviews were a learning

experience on both sides. The program staff benefited from a review of the CHARM proposed integration capabilities, while the IAs gained an understanding of each program's function, data collection process, and how CHARM could bring beneficial opportunities through data sharing and integration.

The analysts documented the programs' processes and where data sharing with other programs would be needed. The analysts documented what data the programs would need to share with the other programs and in what way CHARM would need to support them.

3. Gathering data and developing data dictionaries

During this pass, existing logical models of the programs' data systems were retrieved and reviewed while new logical models were created where none were available. This was one of the most time-consuming activities as most programs were able to provide data only in the form of a technical layout, or coding instructions.

Data models were very important visual tools to communicate how the data supported the programs' business processes and their needs for data sharing. However, the analysts put in considerable effort into adding data definitions – it made sharing this information with the programs much easier. For each program, information was gathered and tabulated in order to define and describe each data element, map it to its source system name, and associate it with a meaningful data group, such as an entity, domain, or topic.

4. Identification of the programs' data sharing needs and limitations

As CHARM is actually a data integration project, most of the requirements deal with specific types of data elements needed, as well as with data sharing issues. During this pass, the analysts met frequently with programs and used the "data dictionaries" to examine the data together and to discuss it in broad sections, as well as in detail, where questions arose. This was more efficient than having the programs review pages of data dictionaries.

During these sessions, the analysts identified and documented:

- the specific data each program was able to share with others,
- any restrictions associated with that sharing,
- any beneficial uses they thought other programs might derive from having access to that data.

In addition, they also identified and documented:

- the data each program wanted to have access to,
- the sources of that data,
- what they thought they could do with that data.

Supporting or restricting legislation and federal laws and policies were reviewed and included where applicable.

5. Clarification of data rules, authority, and default options

Further discussions were conducted in order to decide which data source held, traditionally, the most accurate, the second most accurate, etc. version of shareable data. In cases where the programs collected the same data, this analysis identified the “pecking order” in which the sources will be considered when data is needed.

6. Development of data models for the base data and for the shareable data

The data dictionaries with all the shareable data finalized were used to produce data models for the base data, data models of the shareable data, as well as a CHARM data catalogue. They were also used to identify inconsistent methods of collecting data which, in turn, will be used to assess data standardization needs.

During these activities, the technical developers supplemented the data modeling efforts of the analysts. Technical resources associated with some of the programs also assisted in the development of some of the related data models while the data modeling training provided during the preparatory activities was put to good use by all.

7. Development of data sharing agreements

Once the shareable data models and data dictionaries were developed, the analysts focused on “data sharing agreements” to formalize the data sharing content and conditions specified by the programs.

The development of data sharing agreements started with Vital Records data and was based on a model frequently used to share data among UDOH programs. Even though data sharing fits well with the Public Health purpose of Vital Records, there were still significant concerns as individually identifiable health data were to be shared. By documenting appropriate use legislation and policy, and with all participating programs being internal to the UDOH, the Director of the Office of Vital Records was able to justify sharing the necessary information.

Somewhat daunted by the rigor of the task, the CHARM project team proposed two new approaches. Firstly, they initiated the development and adoption of a Data Stewardship policy. In tandem with a Confidentiality Agreement to be signed by each employee, the new policy encourages data sharing among the Department’s programs in order to optimize data use and to maximize the value of the data as an asset. The Data Stewardship policy will not require Data Sharing Agreements between UDOH programs although it will recommend them for tracking and documentation purposes.

Secondly, the team proposed that each of the five programs only develop data sharing agreements with the CHARM program. This would eliminate the need for each program to establish a data sharing agreement with each of the other four, thus simplifying the process by a factor of two. The template created for VR will be adapted to work with the new Data Stewardship Policy

and with the CHARM program, and will be replicated for all programs involved.

In later CHARM releases, when outside users, such as private providers, will be brought in, agreements for confidentiality and appropriate use will be needed.

8. Identification of each program's service offering and service needs

Although CHARM requirements deal mainly with data, services also need to be identified. The program staff repeatedly noted that they would have a hard time using and correctly interpreting the raw data from another program. Services, are functions that would be performed in the background, would use the shared data "in the raw" and would present it to the requesting programs in a much more usable format. Services are requested by one program and fulfilled by another.

Because of the iterative nature of the development of CHARM, as of the writing of this paper, this activity has not been completed.

This round of sessions will identify what services each program will be able to offer, as part of a CHARM service inventory, as well as what services each program would like to use from the service inventory. Each service will include a definition of the data/service source, data used, transformations performed, results/outputs provided, as well as any limitations as to who might be able to use the service.

Prioritization Activities

Prioritization per se was not needed on the initial set of CHARM requirements. Since integration and data sharing are still new concepts, the programs did not identify so many requirements that a prioritization was needed. It is, however, expected that as the programs accept the CHARM project and embrace its benefits, there will be additional requirements identified. Since CHARM is pursuing an iterative & incremental approach, evolution of the requirements is expected. When new requirements emerge, they will be appropriately addressed and incorporated.

From another perspective, the CHARM sponsors already performed an intrinsic prioritization when they identified a number of releases for the development of CHARM. This white paper only deals with requirements for the first release, which includes five programs, all internal to the UDOH. Functionality related to later releases, such as private providers' requirements, etc., are yet to be identified and documented.

Validation Activities

Validation was immediate in most cases, with the IAs sharing the interview and meeting notes with the program staff soon after each session and asking for verification and validation of content.

All the models were appropriately reviewed and shared with the program staff for content and with the technical team for accuracy and format. In addition, the session involving working with data models and data dictionaries were paced appropriately so that the program staff understood them and were clear as to what data they were identifying to be shared.

Each program performed formal validation by reviewing and approving the documentation for their program. Below is the format used to record each program's requirements:

- A. Program Description and Roles,
- B. How CHARM Will Benefit the Program,
- C. Description of the Program's Data System(s),
- D. Description and Illustration of the Business Processes, and
- E. Data Sharing Issues and Supporting Policy and/or Legislation.

Following program review and approval, these were reviewed in group sessions with the CHARM Core Council, approved and then posted on the project's web site. The CHARM Core Council is composed of the managers of the programs participating in the CHARM integration effort.

A requirements document was produced to formalize all requirements across the programs. The document follows roughly a functional organization format.

CHARM Requirements Document

CHARM Requirements Document was completed and validated. Most of the detail requirements collected are still in notes and in rough documents posted on the project web site.

Sourcing/Authoring

Given the organization of the projects detailed requirements as rough notes, they have retained the authoring information. Because their content was reviewed and approved by the authoring programs, they are a valid reference for requirements sourcing.

Dating

Because all the rough notes are in the form of minutes or interview notes, they were all contemporaneously dated.

Change Management

No changes have yet been made to the original CHARM requirements document.

Versioning

Although CHARM uses an iterative & incremental approach and its requirements are expected to evolve, the current requirements document is at its first version and has not yet gone through a change/revision. Given CHARM's continuous resource stress, the CHARM project team will have to make an extra effort in order to keep the requirements document updated on a timely basis. On the other hand, because of the geographical dispersion of its team, CHARM will have to ensure evolving requirements are adequately captured.

Publishing

The requirements document for CHARM was published as a MS Word document and was shared with the project team and the CHARM Core Council. Because of their concise format, the system's high level requirements have been shared and used in many presentations at the state and national level.

Referencing for Traceability

The CHARM requirements document follows a content-based organization and uses a consistent numbering scheme. As such, it could be the basis for back referencing the various design components and functionality in order to validate for complete coverage. However, during the current project iterations, the CHARM architectural proofs of concept, the team is still referencing the high level requirements. In the next development phases, as well as during testing, the requirements document will be back referenced for traceability.

Acronyms

ADA	American with Disabilities Act
AVR	Active Voice Recognition
CFHS	Community and Family Health Services
CHARM	Child Health Advanced Records Management
CHP	Child Health Profile
CSHCN	Children with Special Health Care Needs
EI	Early Intervention
IA	Information Analyst
JAD	Joint Application Design (sometimes: Joint Application Development)
NCHAM	National Center for Hearing Assessment and Management
NIP	National Immunization Program
RAD	Rapid Application Development
SME	Subject Matter Expert
TQM	Total Quality Management
UDOH	Utah Department of Health
USIIS	Utah Statewide Immunization Information System
USU	Utah State University
VR	Vital Records